ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

Кафедра
«Криптология и кибербезопасность»

Лабораторная работа №6

по предмету «Технологии контейнеризации»

Выполнил студент группы Б20-505

Сорочан Илья

Москва – 2023

Содержание

1. Конфигурация виртуальных машин

```
18  config.vm.define "manager1" do |mgr1|
19    mgr1.vm.network "private_network", ip: "172.20.0.10"
20    mgr1.vm.hostname = "mgr1.local"
21    mgr1.vm.provision :docker
22
23    mgr.vm.provider "virtualbox" do |vb|
24      vb.memory = "1024"
25      vb.cpus = 1
26      vb.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
27    end
28  end
29
30  config.vm.define "worker1" do |wrk1|
31    wrk1.vm.network "private_network", ip: "172.20.0.11"
32    wrk1.vm.hostname = "wrk1.local"
33    wrk1.vm.provision :docker
34
35    mgr.vm.provider "virtualbox" do |vb|
36      vb.memory = "1024"
37      vb.cpus = 1
38      vb.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
39    end
40  end
41
42  config.vm.define "worker2" do |wrk2|
43    wrk1.vm.network "private_network", ip: "172.20.0.12"
44    wrk1.vm.hostname = "wrk2.local"
45    wrk1.vm.provision :docker
46
47    mgr.vm.provider "virtualbox" do |vb|
48      vb.memory = "1024"
49      vb.cpus = 1
50      vb.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
51    end
52  end
```

Рис. 1. Настройка виртуальных машин

2. Инициализация роя

```
vagrant@mgr1:~$ sudo docker swarm init --advertise-addr 172.20.0.10
Swarm initialized: current node (jxx7bheet8g5hvoifoak83396) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-5mladwtzds4kywanozgnvkjkz5dn9m8f1qtu7f9m86p8xqznr7-3y2s7bihf9as7xu7sy5mlb4bn 172.
20.0.10:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

Рис. 2. Инициализация роя на машине manager1

```
vagrant@mgr1:~$ sudo docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-5mladwtzds4kywanozgnvkjkz5dn9m8f1qtu7f9m86p8xqznr7-3y2s7bihf9as7xu7sy5mlb4bn 172.
20.0.10:2377
```

Рис. 3. Получение команды для добавления машины в рой

3. Добавление нод

```
vagrant@wrk1:~$ docker swarm join --token SWMTKN-1-5mladwtzds4kywanozgnvkjkz5dn9m8f1qtu7f9m86p8xqznr7-3y2s7bihf9as7xu7sy
5mlb4bn 172.20.0.10:2377
This node joined a swarm as a worker.
```

Рис. 4. Добавление worker1 в качестве ноды

```
vagrant@wrk2:~$ docker swarm join --token SWMTKN-1-5mladwtzds4kywanozgnvkjkz5dn9m8f1qtu7f9m86p8xqznr7-3y2s7bihf9as7xu7sy
5mlb4bn 172.20.0.10:2377
This node joined a swarm as a worker.
```

Рис. 5. Добавление worker2 в качестве ноды

```
vagrant@mgr1:~$ sudo docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
jxx7bheet8g5hvoifoak83396 *   mgr1       Ready     Active         Leader           24.0.7
bsiigkxndy56c3ojhpjrz2z19     wrk1       Ready     Active                          24.0.7
azqtxad3koxjym7uch941ppht     wrk2       Ready     Active                          24.0.7
```

Рис. 6. Список нод

```
 4   $manager_script = <<SCRIPT
 5   sudo docker swarm init --advertise-addr 172.20.0.10
 6   sudo docker swarm join-token --quiet worker > /vagrant/worker_token
 7   SCRIPT
 8
 9   $worker_script = <<SCRIPT
10   sudo docker swarm join --token $(cat /vagrant/worker_token) 172.20.0.10
11   SCRIPT
```

Рис. 7. Vagrant script

4. Развертывание сервиса в кластере

```
vagrant@mgr1:~$ sudo docker service create --replicas 1 --name helloworld alpine ping docker.com
pgd1yns23qfmv9rvhdk1wt0cv
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Service converged
vagrant@mgr1:~$ sudo docker service ls
ID             NAME         MODE         REPLICAS   IMAGE          PORTS
pgd1yns23qfm   helloworld   replicated   1/1        alpine:latest
```

Рис. 8. Запуск hello world на кластере

- команда `docker service create` создает службу;
- флаг `--name` называет службу helloworld;
- флаг `--replicas` указывает желаемое состояние одного работающего экземпляра;
- аргументы `alpine ping` docker.com определяют службу как контейнер Alpine Linux, который выполняет команду `ping docker.com`.

5. Инспектирование сервиса в кластере

```
vagrant@mgr1:~$ sudo docker service inspect --pretty helloworld

ID:             pgd1yns23qfmv9rvhdk1wt0cv
Name:           helloworld
Service Mode:   Replicated
 Replicas:      1
Placement:
UpdateConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order:    stop-first
ContainerSpec:
 Image:         alpine:latest@sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851978
 Args:          ping docker.com
 Init:          false
Resources:
Endpoint Mode:  vip
```

Рис. 9. Просмотр информации о контейнере

Вывод команды `sudo docker service inspect helloworld`:

```
[
    {
        "ID": "pgd1yns23qfmv9rvhdk1wt0cv",
        "Version": {
```

```
            "Index": 21
        },
        "CreatedAt": "2023-10-31T17:36:51.670589461Z",
        "UpdatedAt": "2023-10-31T17:36:51.670589461Z",
        "Spec": {
            "Name": "helloworld",
            "Labels": {},
            "TaskTemplate": {
                "ContainerSpec": {
                 "Image": "alpine:latest@sha256:eece025e432126ce23f223450a0326f
                    "Args": [
                        "ping",
                        "docker.com"
                    ],
                    "Init": false,
                    "StopGracePeriod": 10000000000,
                    "DNSConfig": {},
                    "Isolation": "default"
                },
                "Resources": {
                    "Limits": {},
                    "Reservations": {}
                },
                "RestartPolicy": {
                    "Condition": "any",
                    "Delay": 5000000000,
                    "MaxAttempts": 0
                },
                "Placement": {
                    "Platforms": [
                        {
                            "Architecture": "amd64",
                            "OS": "linux"
                        },
                        {
                            "OS": "linux"
```

```
                    },
                    {
                        "OS": "linux"
                    },
                    {
                        "Architecture": "arm64",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "386",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "ppc64le",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "s390x",
                        "OS": "linux"
                    }
                ]
            },
            "ForceUpdate": 0,
            "Runtime": "container"
        },
        "Mode": {
            "Replicated": {
                "Replicas": 1
            }
        },
        "UpdateConfig": {
            "Parallelism": 1,
            "FailureAction": "pause",
            "Monitor": 5000000000,
            "MaxFailureRatio": 0,
            "Order": "stop-first"
```

```
            },
            "RollbackConfig": {
                "Parallelism": 1,
                "FailureAction": "pause",
                "Monitor": 5000000000,
                "MaxFailureRatio": 0,
                "Order": "stop-first"
            },
            "EndpointSpec": {
                "Mode": "vip"
            }
        },
        "Endpoint": {
            "Spec": {}
        }
    }
]
```


```
vagrant@mgr1:~$ sudo docker service ps helloworld
ID             NAME           IMAGE          NODE      DESIRED STATE   CURRENT STATE           ERROR     PORTS
tgo72m3wp570   helloworld.1   alpine:latest  mgr1      Running         Running 8 minutes ago
```

Рис. 10. На каких нодах запущен сервис

6. Масштабирование сервера в кластере

```
vagrant@mgr1:~$ sudo docker service scale helloworld=5
helloworld scaled to 5
overall progress: 5 out of 5 tasks
1/5: running   [==================================================>]
2/5: running   [==================================================>]
3/5: running   [==================================================>]
4/5: running   [==================================================>]
5/5: running   [==================================================>]
verify: Service converged
vagrant@mgr1:~$ sudo docker service ps helloworld
ID             NAME           IMAGE          NODE      DESIRED STATE   CURRENT STATE            ERROR     PORTS
tgo72m3wp570   helloworld.1   alpine:latest  mgr1      Running         Running 10 minutes ago
pvuk0gjet5a3   helloworld.2   alpine:latest  wrk1      Running         Running 14 seconds ago
8pa527iov208   helloworld.3   alpine:latest  wrk1      Running         Running 14 seconds ago
n73u16od54sw   helloworld.4   alpine:latest  mgr1      Running         Running 22 seconds ago
3gi9fmo5ra0b   helloworld.5   alpine:latest  wrk2      Running         Running 14 seconds ago
```

Рис. 11. Масштабирование

7. Удаление сервиса

```
vagrant@mgr1:~$ sudo docker service rm helloworld
helloworld
vagrant@mgr1:~$ sudo docker service inspect helloworld
[]
Status: Error: no such service: helloworld, Code: 1
```

Рис. 12. Удаление контейнера

Проверим как быстро были удалены контейнеры на разных машинах:

```
vagrant@mgr1:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Рис. 13. `docker ps` на manager1

```
vagrant@wrk1:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Рис. 14. `docker ps` на worker1

```
vagrant@wrk2:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Рис. 15. `docker ps` на worker2

8. Обновление сервиса

```
vagrant@mgr1:~$ sudo docker service create --replicas 3 --name redis --update-delay 10s redis:3.0.6
brdvkchg680gupe1adr3dzeph
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
```

Рис. 16. Создание redis контейнера

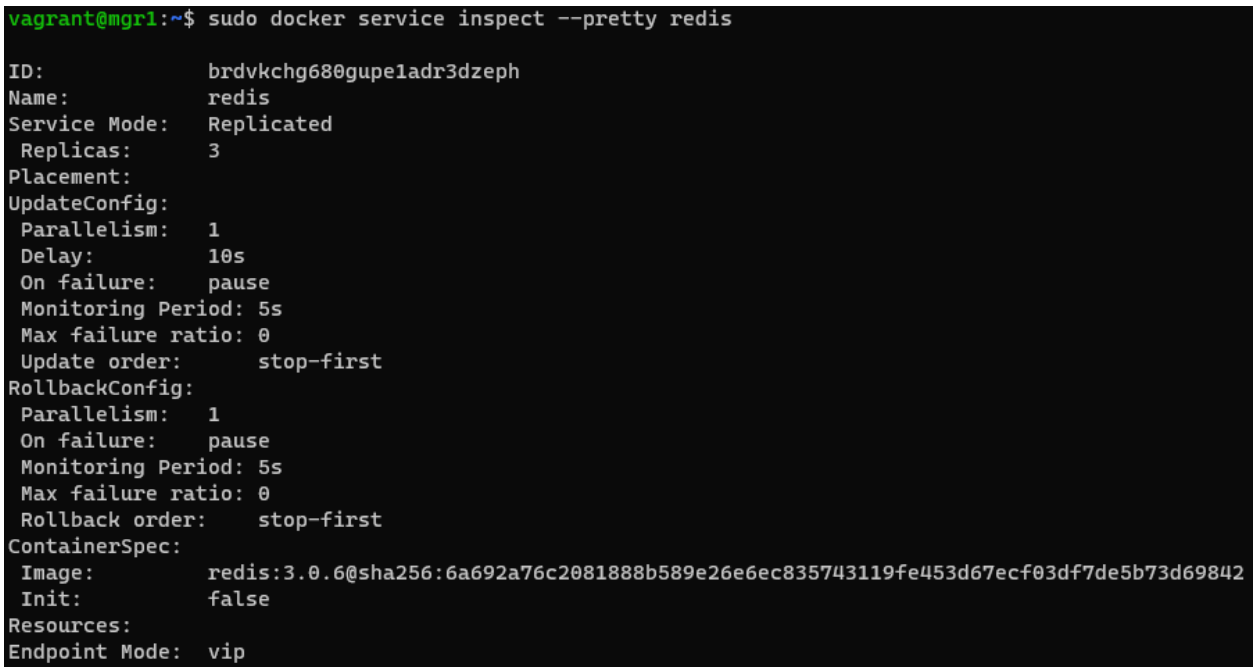

Рис. 17. Информация о контейнере

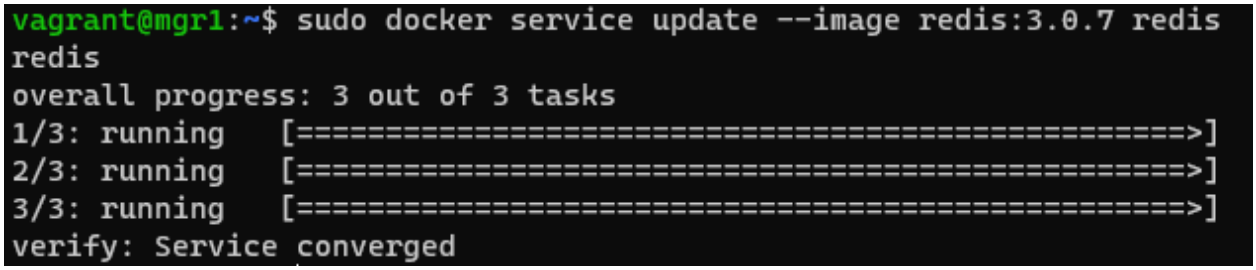

Рис. 18. Обновление

- остановка первого task;
- планирование обновления для остановленной задачи (task);
- запуск контейнера обновленной задачи;
  - если обновление задачи возвращает RUNNING, ожидание установленной задержки, а затем запуск задачи;
  - если в процессе обновления задача возвращает FAILED, остановка

обновления.

```
vagrant@mgr1:~$ sudo docker service inspect --pretty redis

ID:             brdvkchg680gupe1adr3dzeph
Name:           redis
Service Mode:   Replicated
 Replicas:      3
Placement:
UpdateConfig:
 Parallelism:   1
 Delay:         10s
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order:    stop-first
ContainerSpec:
 Image:         redis:3.0.7@sha256:730b765df9fe96af414da64a2b67f3a5f70b8fd13a31e5096fee4807ed802e20
 Init:          false
Resources:
Endpoint Mode:  vip
```

Рис. 19. После обновления

```
vagrant@mgr1:~$ sudo docker service update redis
redis
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
```

Рис. 20. Запуск обновленной версии

```
vagrant@mgr1:~$ sudo docker service ps redis
ID             NAME          IMAGE         NODE      DESIRED STATE   CURRENT STATE            ERROR                              PORTS
w9aywzmxr56d   redis.1       redis:3.0.7   mgr1      Running         Running 5 minutes ago
pjtfyrybdgit    \_ redis.1   redis:3.0.7   mgr1      Shutdown        Rejected 5 minutes ago   "No such image: redis:3.0.7@sh…"
q22mp1u561p4    \_ redis.1   redis:3.0.7   wrk1      Shutdown        Shutdown 5 minutes ago
tw0j7qa7pkis    \_ redis.1   redis:3.0.6   mgr1      Shutdown        Shutdown 12 minutes ago
e27t4d8qp1mc    \_ redis.1   redis:3.0.6   wrk2      Shutdown        Running 14 minutes ago
vkbwc5z2yen2   redis.2       redis:3.0.7   mgr1      Running         Running 5 minutes ago
7p4im3yfkvc1    \_ redis.2   redis:3.0.7   wrk1      Shutdown        Shutdown 5 minutes ago
qzgzjhr21p6g    \_ redis.2   redis:3.0.6   mgr1      Shutdown        Shutdown 5 minutes ago
iam6vvgqffk3   redis.3       redis:3.0.7   mgr1      Running         Running 5 minutes ago
g633vu1mhfnz    \_ redis.3   redis:3.0.7   wrk1      Shutdown        Shutdown 5 minutes ago
oiqvrgc3wbk2    \_ redis.3   redis:3.0.7   mgr1      Shutdown        Shutdown 5 minutes ago
qmsjsvwbx90b    \_ redis.3   redis:3.0.6   wrk1      Shutdown        Shutdown 11 minutes ago
```

Рис. 21. Контейнеры

9. Обслуживание нод кластера

На предыдущих этапах руководства все узлы работали в ACTIVE режиме. Менеджер кластера может назначать задачи любому ACTIVE узлу, поэтому до сих пор все узлы были доступны для получения задач. Иногда, например, во время планового обслуживания необходимо установить узел в режим DRAIN.

DRAIN режим не позволяет узлу получать новые задачи от менеджера кластера. Это также означает, что менеджер останавливает задачи, выполняемые на узле, и запускает задачи реплики на ACTIVE доступном узле.

```
vagrant@mgr1:~$ sudo docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
jxx7bheet8g5hvoifoak83396 *   mgr1       Ready     Active         Leader           24.0.7
bsiigkxndy56c3ojhpjrz2z19     wrk1       Ready     Active                          24.0.7
azqtxad3koxjym7uch941ppht     wrk2       Down      Active                          24.0.7
```

Рис. 22. Проверка доступности нод

```
vagrant@mgr1:~$ sudo docker service create --replicas 3 --name redis --update-delay 10s redis:3.0.6
gsmvqykp5q9olkrvi5dsh24g0
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
```

Рис. 23. Создание сервиса

```
vagrant@mgr1:~$ sudo docker node update --availability drain wrk1
wrk1
vagrant@mgr1:~$ sudo docker node inspect --pretty wrk1
ID:                     bsiigkxndy56c3ojhpjrz2z19
Hostname:               wrk1
Joined at:              2023-10-31 17:22:19.849891942 +0000 utc
Status:
 State:                 Ready
 Availability:          Drain
 Address:               172.20.0.11
Platform:
 Operating System:      linux
 Architecture:          x86_64
Resources:
 CPUs:                  2
 Memory:                957.3MiB
Plugins:
 Log:           awslogs, fluentd, gcplogs, gelf, journald, json-file, local, logentries, splunk, syslog
 Network:               bridge, host, ipvlan, macvlan, null, overlay
 Volume:                local
Engine Version:         24.0.7
TLS Info:
 TrustRoot:
-----BEGIN CERTIFICATE-----
MIIBajCCARCgAwIBAgIUYVsjOWEq7bh8gCOspYy+5yct8sswCgYIKoZIzj0EAwIw
EzERMA8GA1UEAxMIc3dhcm0tY2EwHhcNMjMxMDMxMTcxNDAwWhcNNDMxMDI2MTcx
NDAwWjATMREwDwYDVQQDEwhzd2FybS1jYTBZMBMGByqGSM49AgEGCCqGSM49AwEH
A0IABIWluMKnoufGDtyrlsIcSmps3Y9DEnz+1ndUsH2mI2aob1ebqWGyjrl5yBvw
HyNnxlPPToY+HNWMO+2y+yFObc+jQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMB
Af8EBTADAQH/MB0GA1UdDgQWBBQa7QX7c7OwIYFlxSXMgXBs8GlqVzAKBggqhkjO
PQQDAgNIADBFAiAus5P1Ge3o10xZIBU4suKR+ihv68S/5I+HkJBp1ok9LwIhAIPU
CiRcX79xYEiwAXcRU8Vpsf7nQKI3vyWnlRvbzyXB
-----END CERTIFICATE-----

 Issuer Subject:        MBMxETAPBgNVBAMTCHN3YXJtLWNh
 Issuer Public Key:     MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEhaW4wqei58YO3KuWwhxKamzdj0MSfP7Wd1SwfaYjZqhvV5upYbKOuXnIG/AfI2fGU89Ohj4c1Yw77bL7I
U5tzw==
```

Рис. 24. Перевод worker1 в режим DRAIN

```
vagrant@mgr1:~$ sudo docker service ps redis
ID             NAME          IMAGE          NODE      DESIRED STATE   CURRENT STATE                ERROR     PORTS
zhnqi4z7r9e1   redis.1       redis:3.0.6    mgr1      Running         Running 2 minutes ago
y4og067lqmd7   redis.2       redis:3.0.6    mgr1      Running         Running about a minute ago
xjpv7a77j3pj    \_ redis.2   redis:3.0.6    wrk1      Shutdown        Shutdown about a minute ago
tp5gowh2pbpy   redis.3       redis:3.0.6    mgr1      Running         Running about a minute ago
vu9sn7nrni5t    \_ redis.3   redis:3.0.6    wrk1      Shutdown        Shutdown about a minute ago
```

Рис. 25. ps после перевода worker1 в DRAIN

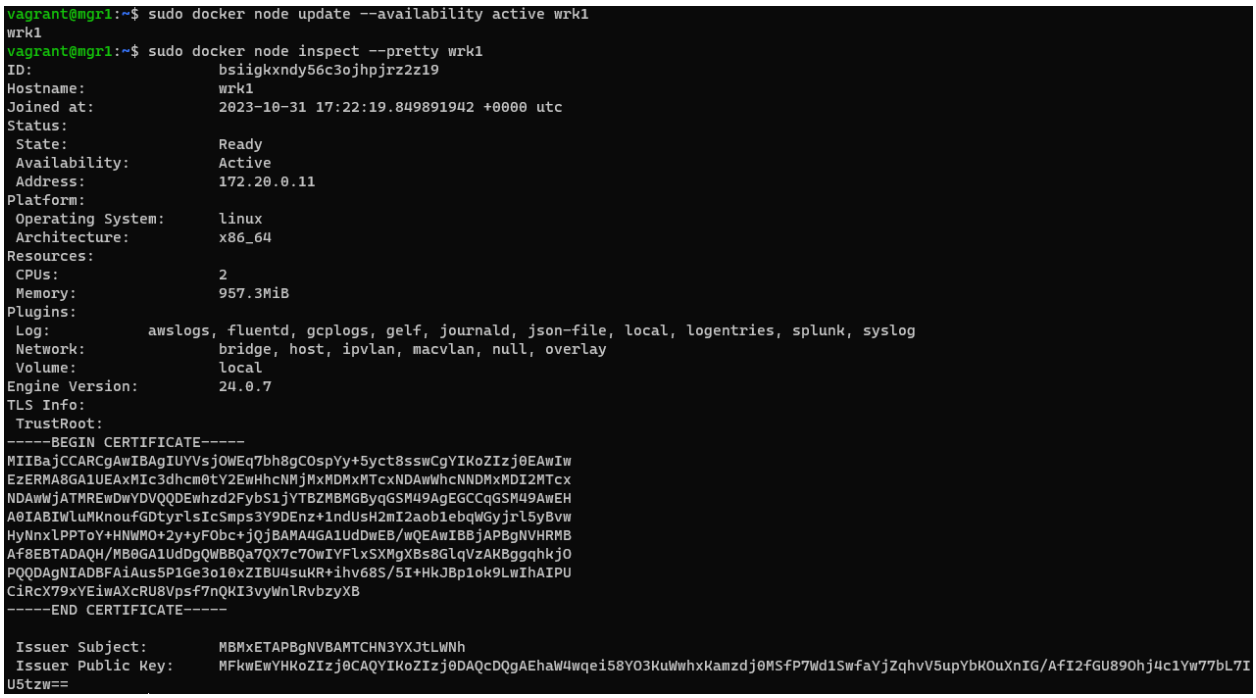

Рис. 26. Перевод worker1 обратно в режим AVALIABLE